ARTICLE FOR PEER-REVIEW

Multidisciplinary Design Optimization of a Generic B-Pillar under Package and Design Constraints

ARTICLE HISTORY
Compiled September 3, 2020

ABSTRACT
This article introduces a novel constraining approach for structural optimization, which aims to support the conceptual engineer during the early embodiment phase for structural lightweight design. It reduces the time spent on structural engineering studies by enabling optimization algorithms to detect geometric intersections by analyzing the mesh information. This article reviews approaches from the literature focusing on CAD-environments, sampling methods, data analytics and optimization techniques for design and sizing optimization with FE-models. The evaluated approaches are integrated into a Python-based optimization environment. Accordingly, the introduced methodology enables the environment to handle geometric infeasible designs. The presentation of the first results focuses on the feasibility of structural assemblies and the results demonstrate the viability of the NSGA-II for optimization tasks. The example considers the design of a generic b-pillar structure under crash-safety requirements. Using this approach, the NSGA-II algorithm avoids geometric infeasible areas and comparably increases structural performance.

KEYWORDS
multidisciplinary design optimization; constraint strategies; lightweight design; geometric feasibility

1. Introduction

Nowadays, product turnover times are steadily shortening. Furthermore, the structural complexity of systems is increasing, resulting in a need for new tools and optimization techniques. To be capable of the challenges in the product development process (PDP), engineers are often using comparative studies during the conceptual design phase. Before performing structural analysis, an evaluation with a cost-benefit-analysis helps to reduce the number of initial designs. Furthermore, methodical and analytical methods for the conceptual design phase exist, e.g. presented by Fröhlich et al. (2017). In the subsequent embodiment design phase, finite-element (FE) analysis helps to gain a more detailed knowledge of the product during the early embodiment and the final detail design. The requirements of each product lead to different formulations of objectives and constraints, which can be used to deploy multidisciplinary design optimizations (MDOs). MDOs help to find the best structural design by an automated structural optimization process. An example of such an automated process chain for numerical analysis under analytical constraints is presented by Ghaffarimejlej, Türck, and Vietor (2016). The use of such computational automation and optimization for numeric simulations can accelerate the embodiment process by considering multiple objectives and constraints. Exemplarily, cost, manufacturing, weight and crash constraints are mentioned. These automated processes are not able to detect and handle external ge-

https://doi.org/10.24355/dbbs.084-202111091534-0

ometric constraints, which are mostly a result of packaging or design requirements. Therefore, this article introduces an approach for the detection and handling of geometric infeasibility in MDO routines.

Hence, Section 2 gives an overview of the topic of shape and design optimization in structural design. The presented main topics are parametric CAD modeling, DoE methods, metamodeling and optimization techniques. Section 3 introduces a feasibility assessment approach for multidisciplinary design optimization. A geometric factor is introduced, which is applied as the infeasibility metric. Section 4 shows the capability of the approach in combination with an NSGA-II optimization routine, as different optimization studies are performed for a generic b-pillar. Accordingly, a discussion of the results takes place in Section 5. The final section summarizes the results and points out further areas of activity and subsequent steps.

2. Materials and Methods

The following chapter briefly introduces the engineering design process and classifies optimization approaches in the design process. Secondly, the method of parametric engineering design is introduced. Metamodeling and surrogate-modeling approaches are stated in the third section. Finally, an overview of the theoretical foundations and examples for engineering shape and sizing optimization is given.

2.1. *The Engineering Design Process*

The engineering design process guides the engineer through the development of new products. In general, the development process of a product can be split into two parts, the development and the production phase. The main steps of the product development process are stated by Feldhusen and Grote (2013); namely as the planning, the design process (subdivided into conceptual-, embodiment- and detailed design), the documentation, and production. After finalization, the product should fulfil multiple requirements. To sustain these requirements, the design engineer needs to consider all influencing factors sufficiently. Ideally, all requirements form evaluable criteria that need to be fulfilled, under consideration of all influencing factors. A perfect product development process would need ideal product knowledge. In a realistic product development process, the real product knowledge will always be of minor depth than ideal product knowledge. Thus, the foundation for many design decisions is based on assumptions, estimations and experience. To increase the quality of the final design, the real product knowledge in conceptual, embodiment, and detail design should be increased. They are the most vital factors, to ensure well-founded decisions in the earliest possible phase, Feldhusen and Grote (2013). The fulfilment of multiple requirements of every structural component needs to be considered during the automotive design process. Friedrich (2013) states the main requirements during the automotive design process for large-scale lightweight products as suitability for series production. These are primary profitability, lightweight design, stiffness, strength, packaging, crash performance, NVH, recycling, repair costs, and surface quality. The assessment of all these requirements should be done as early as possible in the product development process (PDP), to avoid expensive subsequent adjustments in the later development process. The use of methods and tools supports a steady process. For the embodiment phase, different optimization strategies are applicable, Schumacher et al. (2005). After an initial choice of conceptual design, suitable materials should be selected. Finally,

https://doi.org/10.24355/dbbs.084-202111091534-0

the structural design follows. Three classes of structural optimization problems exist, which depend on the concept variables and are namely:

- Topology Optimization
- Shape Optimization
- Sizing Optimization

As topology optimization focusses on the position and arrangement of the structural elements, shape optimization changes the geometry, while keeping the basic design elements. Finally, sizing optimization (also referred to as size optimization) varies the thickness and cross-section parameters, keeping all other parameters unchanged.

Shape optimization approaches use parametric modeling techniques. Furthermore, the shape optimization methods are split into node-based (referred to as morphing) and CAD-based shape optimization, see Bletzinger (2018). Node-based approaches are normally limited in their maximum displacement, as the stretching of elements results in poor mesh-quality. A low mesh-quality causes the numerical quality to be lower. Thus, node-based shape optimization approaches have their scope on simpler geometric optimization tasks. Conversely, CAD-based parametrization techniques exist, where CAD models are exported to a pre-processor, which creates the FE-model. This kind of approach is classified as shape optimization. This kind of approach requires the linkage of CAD and FE-methods, which often leads to errors and laborious processes. For better geometric design and mesh quality, the automated FE-meshing is striven, e.g. demonstrated by a NURBS-based approach by Winter et al. (2019). Hence, another approach to handle this process is the topology-based, implicit CAD structure, e.g. described by Zimmer (2002).

2.2. *Parametric CAD-Modeling for Design Optimization*

Simple parametric designs can be realized in nearly all CAD-environments. To avoid complicated models and error-prone processes, special tools exist, where the issues with automated parametric designs can be handled. In SFE CONCEPT (Dassault Systèmes 2020a) , the key feature is the implicit parametric CAD-environment, which enables point, line, and surface-based parametric models. Joining technologies such as welding or adhesive bonding define the topological interaction of the basic elements. By changing the spatial orientation, geometrical changes are implied, as shown in the introductive paper of Zimmer (2006). Besides, these topological changes may be recorded and can be used as Design Variables (DVs) for shape and size optimization, which is demonstrated in the work of Duddeck (2016). Accordingly, the internal pre-processor of SFE CONCEPT can create FE-meshes from a surface representation and export these meshes for numerical calculations. With its parametrization and meshing ability, SFE CONCEPT shortens the conceptual product development process for structural parts and assemblies. Exemplarily, Rayamajhi, Hunkeler, and Duddeck (2014) give a detailed introduction into the implicit parametrization technique used in SFE CONCEPT. For a better understanding of geometric parametrizations, Figure 1 shows a generic b-pillar with implicit parametrization for two possible designs.

Ideally, realizing implicit designs helps to prevent surface penetrations during the automated design of the pillar structure. The structural potential of the concept may be further exploited, as optimization algorithms are coupled with the parametric FE-environment of SFE CONCEPT by batch commands, Hilmann (2011). Different conditions should be fulfilled by the DVs to ensure a clear trending of the influencing pa-

https://doi.org/10.24355/dbbs.084-202111091534-0

Figure 1. Implicit parametrization of a generic b-pillar. Left-hand side in original shape with oblique view. The middle picture shows the maximum DV configuration and right-hand side the minimum DV configuration. The thickness of the b-pillar is changed in the red cross-sections

rameters and fast convergence of the optimization algorithm. Duddeck (2016) names the utmost important assumptions and conditions for a suitable choice of DVs and further measures, for example:

- the number of DVs should be kept as small as necessary to enable a sound optimization procedure and should be large enough to realize all relevant geometrical changes
- a hierarchic approach should be used, to link sensitivities of DVs directly to the objectives and constraints
- such sensitivities should apply on the full design space, not relying on local crash behavior
- the responses need to be comparable between different stages in product development and between different types of the product

Accordingly, the right choice of the DVs may help to reduce the time for prediction and optimization. For such predictions and to reduce the cost-expensive FE-simulations, metamodels and surrogate models can be used, as stated by Ryberg, Domeij Bäckryd, and Nilsson (2012). As well, surrogate models are necessary to enable faster optimization procedures.

2.3. *Metamodeling and Surrogate Modeling*

By the automated parameterization and calculation of CAE-models, a widespread of variants can be evaluated by approaches such as the Design of Experiments (DoE). Therefore, multiple DVs are set by a given pattern to achieve a wide and representative range of responses for the given solution space. Searching the best solution for the given design-range, DoE approaches are limited due to the big amount of computing power needed. Especially crash simulations have an extensive need for computing resources. Accordingly, the responses to the DVs should be used to implement mathematical optimization procedures to find the best design result for the objective function of the given model. Consequently, such optimization procedures are the key to a fast and resource-efficient design process. During optimization problems, using mathematical and physical surrogate models can help to reduce the computational effort, especially for crash simulations, (Rayamajhi, Hunkeler, and Duddeck 2015). Sub-structures or linear models with Equivalent Static Load Method (ESLM) can sometimes describe

https://doi.org/10.24355/dbbs.084-202111091534-0

sufficiently well the physical behavior. Furthermore, using mathematical surrogate models for predictions and optimizations helps to reduce the time for predictions even further. Commonly used types are: Polynomial Regression, Radial Basis Function, Kriging or Neural Network approaches, (Bäckryd, Ryberg, and Nilsson 2017; Kiani and Yildiz 2016). Moreover, trending of sensitivities for design variables needs to be considered, especially if many DVs are used. Often during the optimization, only some DVs are essential for the objectives and constraints. Besides, multiple constraints are relevant during crash optimization, as discussed in Schumacher et al. (2005). Examples are the specific force levels of several parts, high-energy absorption with fold buckling or the smooth acceleration time-curve.

Following Paas and van Dijk (2018), the uncertainty on a metamodel should be validated for the final converged result. The uncertainty tends to appear for inappropriate regression techniques, lack of data from the DoE, too many DVs, and numerical uncertainty in the FE-model. An extension of the data by the uncertainty enables the optimization routine to consider the uncertainties during the optimization run. The designs, which fail to fulfil the performance constraints are infeasible, (Rayamajhi, Hunkeler, and Duddeck 2015). Accordingly, the use of penalty functions enables to embed performance constraints in the fitness function.

2.4. *Methods for Multidisciplinary Design Optimization*

In the case of automotive crash safety, metaheuristic optimization approaches are a commonly used "state-of-art" optimization technique. Ordinarily, their ability to explore large solution spaces within a reasonable time qualifies them. In general, an MDO can be defined following (Giesing and Barthelemy 1998, p. 2):

> "A methodology for the design of complex engineering systems and subsystems that coherently exploits the synergism of mutually interacting phenomena."

The aim of such optimization approaches is the minimization (or maximization) of an unknown objective function $f(x)$ to find the optimal design x^* for a given design space χ. The design variables x of the design space need to stay within the lower boundary x_{lower} and the upper boundary x_{upper}. For constrained optimization problems, soft constraints $g(x)$ and hard constraints $h(x)$ need to be considered during optimization, (Ryberg, Domeij Bäckryd, and Nilsson 2012; Shahriari et al. 2016). Equation (1) gives the formulation for a constrained single-objective optimization problem.

$$\begin{aligned} x^* &= \min_{x \in \chi} f(x) \\ subject\ to\ g(x) &\leq 0 \\ h(x) &= 0 \\ x_{lower} \leq x &\leq x_{upper} \end{aligned} \tag{1}$$

The mentioned single objective optimization is incapable to optimize multiple objectives. For the minimization of a problem with more than one objective, a multi-objective optimization (MOO) problem needs to be formulated, see Equation (2).

https://doi.org/10.24355/dbbs.084-202111091534-0

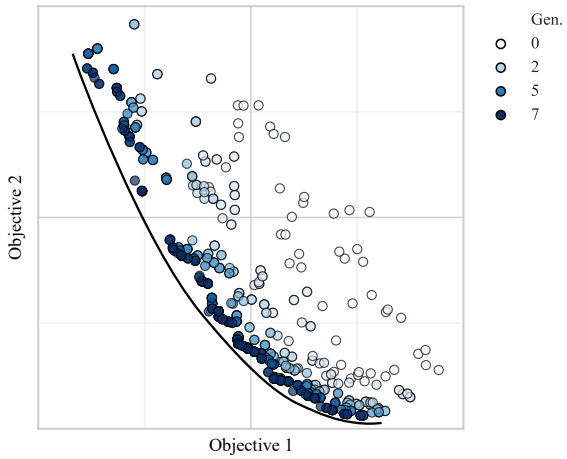


Figure 2. Pareto ideal-design for a Multi-Objective Optimization with two objectives, as a result of multi-disciplinary optimization (NSGA-II algorithm). Dark blue elements form the Pareto-frontier, represented by the black line

$$x^* = \min_{x \in \chi} f_1(x), ..., f_m(x) \quad subject\ to\ g(x) \leq 0 \tag{2}$$

A multi-objective optimization will result in a Pareto-ideal design, as shown in Figure 2. The resulting set of optimized designs in the Pareto frontier represents multiple solutions, which fulfil the criteria of the multi-objective optimization task. Therefore, the designer can choose from a broader set of solutions than a single objective optimization can give, e.g. Ryberg et al. (2012).

2.5. *An Overview of Engineering Design and Sizing Optimization*

The following section gives an overview of state-of-the-art examples for Body in White (BIW) optimization approaches, which focus on crash and Noise-Vibration-Harshness (NVH) optimization. Wang and Cai (2018) implemented a hybrid metaheuristic Particle Swarm Optimization (PSO) algorithm with the Bacterial Foraging Optimization (BFO) to optimize the structural performance of a BIW structure. The performance of the hybrid metaheuristic approach was superior to that of the single algorithms. Thereby, they used a Radial Basis Function (RBF) as a surrogate model. Moreover, precedent crash tests assured the validity of the SFE CONCEPT model of Wang and Cai.

In a comparative study, Kiani and Yildiz (2016) implemented five different metaheuristics. The objectives and constraints resulted from NVH and crashworthiness requirements, using an RBF and miscellaneous algorithms to optimize on that surrogate model. In their study, the Differential Evolution Algorithm (DE) outperformed the Artificial Bee Colony Algorithm (ABC), the Simulated Annealing (SA), the Particle Swarm Optimization (PSO) and the Genetic Algorithm (GA). The DE improved the total weight of the BIW by 3 %, while it fulfilled the constraints. Paas and van Dijk (2018) used the Response Surface Method to build a meta-model of the calculated responses. On this meta-model, they applied an NSGA-II for the optimization for an

https://doi.org/10.24355/dbbs.084-202111091534-0

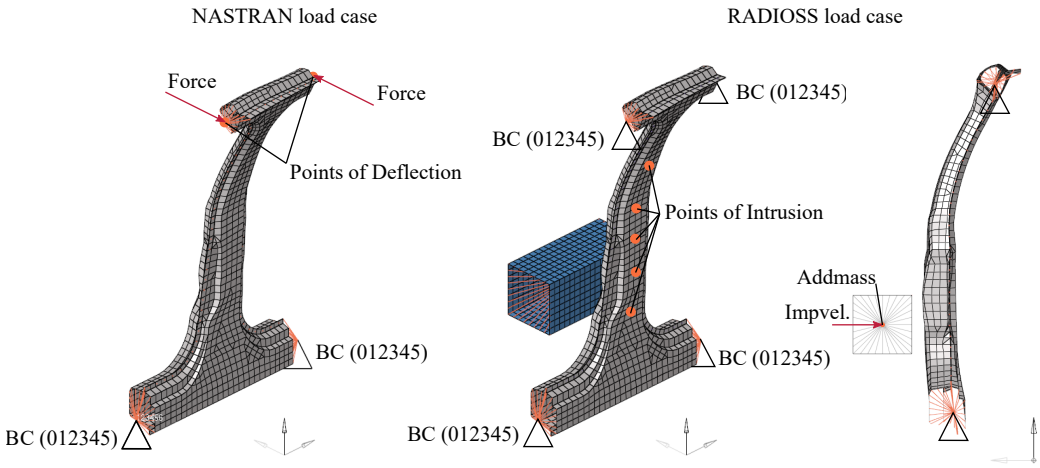


Figure 3. RADIOSS and NASTRAN load case shown for the generic study. The torsional stiffness of the b-pillar is evaluated in NASTRAN, while the intrusion and summed internal energy are calculated in RADIOSS.

inner automotive hood panel. The NSGA-II outperformed alternative metaheuristics for various cross attribute weight optimization tasks. Such standard MDO may involve up to 50 load-cases, 150 DVs, and up to 1000 response functions. Furthermore, Bayes-Optimization approaches are a popular approach to minimize a given objective function. It is a model-based approach, in which a sequential posterior updating refines a prior belief of the objective function. The minimum expected risk approach leads to acquisition functions, which trade-off explorative and exploitative behavior. This makes the Bayesian Optimization very attractive for numerical costly evaluations of the objective function $f(x)$, such as crash optimization, (Shahriari et al. 2016).

3. Fitness Formulation, Feasibility and Assessment of a Generic B-Pillar Structure

This chapter introduces the generic load case, a methodology for feasibility assessment and the handling of feasibility in shape optimization approaches. The first section describes the fitness formulation for the given example. The method for the detection and handling of infeasible designs is introduced in the second section. It also describes the manipulation of the fitness of infeasible designs for the given fitness formulation.

3.1. *Load case, Fitness Formulation and Penalization*

Initially, this section describes the generic load case covered in this article. The optimization considers two load cases for the objectives and constriants, see Figure 3.

A momentum represents the torsional stiffness of the b-pillar around its neutral axis. The NASTRAN load case is created by two forces on the top-end and two fixed bearings at the bottom-end. $Y_{C,torsion}$ represents the torsional-stiffness, which is measured at the top-end force points. The RADIOSS load case serves as the side-impact scenario. An impactor with an

https://doi.org/10.24355/dbbs.084-202111091534-0

added mass of one ton and an imposed velocity of 50 km/h hits the b-pillar in the lateral direction. The maximum intrusion during the whole simulation at 5 points on the inside of the b-pillar is measured. The maximum value from these 5 points is taken for the intrusion $Y_{C,intrusion}$. The summarized internal energy (elastic and plastic) of the whole b-pillar represents the absorbed internal energy of $Y_{C,energy}$. Both values are evaluated under consideration of plasticity. This formulation results in a maximum absorbed energy, while the intrusion into the passenger compartment stays minimal.

The population fitness is calculated from the responses by the evaluation of the normalized objective and the summed normalized constraints. As an objective formulation, a direct measure of a percentage formulation is chosen. The mass will be minimized after the following formulation in Equation (3), which is presenting the implemented objective formulation.

$$Y_{O,mass} = \frac{Y_O}{Y_{O,init}} \tag{3}$$

Therein the mass of an initial part $Y_{O,init}$ is used as a reference. The initial part is calculated by an initial reference combination of the parameters, which might be chosen from preceding designs. The objective value is calculated referring to the initial design, with the current objective value of Y_O. The objective value will be minimized by the algorithm, see Equation (4).

$$\min_{x \in \chi} f_0(x) = Y_{O,mass} \tag{4}$$

For the normalization of the constrained values, a min-max normalization is used. The method is exemplarily described by Messac, Ismail-Yahaya, and Mattson (2003) and rescales the constraints to a normative value. For $Y_{C,min}$ and $Y_{C,max}$ a certain threshold is used.

$$Y_{C,normalized} = \frac{Y_C - Y_{C,init,min}}{Y_{C,init,max} - Y_{C,init,min}} \tag{5}$$

If a constraint is exceeding a certain threshold, the value is penalized. The implementation of the penalization procedure follows the soft constraint introduced in Section 2.4. The constraints are set by reference to the initial design $Y_{C,init}$. If the metrical constraint value of a design Y_C is violating the corresponding performance constraint $Y_{C,init}$, the normalized constraint value $Y_{C,normalized}$ will be penalized. If the metrical constraint value Y_C fullfills the corresponding performance constraint $Y_{C,init}$ the normalized value $Y_{C,normalized}$ is treated as an objective value.

$$|Y_{C,normalized}| = \begin{cases} (Y_{C,normalized})^2 & , Y_C \geq Y_{C,init} \\ Y_{C,normalized} & , \text{else} \end{cases} \tag{6}$$

This approach is used to formulate the fitness function by a weighted sum-method, Equation (7).

https://doi.org/10.24355/dbbs.084-202111091534-0

$$\min_{x \in \chi} f_c(x) = \alpha \cdot Y_{C,torsion} + \beta \cdot \frac{1}{Y_{C,energy}} + \gamma \cdot Y_{C,intrusion} \tag{7}$$

The second objective is the weighted sum of the constrained objectives. The constraints $Y_{C,torsion}$, $Y_{C,energy}$ and $Y_{C,intrusion}$ are penalized if they are violated by the design. Considering the penetrations, the infeasible designs of the sampling will have the worst performance of the sample. For the constraints, different approaches have been used, following a cumulative paper on penalization techniques by Yeniay (2005), namely static penalties. Following, a high fitness value represents bad structural performance, while a low fitness value implements a high structural performance.

3.2. *Feasibility of Assemblies*

The following approach is applicable when package restrictions or other types of infeasible design occur. Complex geometric changes of new assemblies may cause geometric intersections of surfaces, as mentioned in Section 2.2. Therefore, a feasibility assessment is introduced which enables the automated check, and if necessary, the penalization of the infeasible designs. As discussed in Section 2.3 infeasible designs are the ones, which fail to fulfil any kind of performance constraint. Furthermore, these constraints can be hard or soft see Section 2.4. The aim of the following geometric feasibility assessment is the formulation of a soft-constraint, which should be able to differentiate the scale of the infeasibility and not just identify surface penetrations. In the following, infeasibility is always the byword for geometric infeasibility.

3.2.1. Methodical Approach

For every created design of an initial sampling or during any optimization process, the assessment of feasibility is necessary as new designs may exhibit surface penetrations. For instance, the simple exclusion of infeasible designs may cause the optimization to fail to converge. The implementation of a routine, which enables proper handling of infeasibilities, addresses this need. The procedure identifies infeasible designs by an external mesh check in ANSA (beta cae 2020), which processes the analysis with an intersection check. The penalization of the fitness function allows the penalization of infeasible designs by manipulating the fitness value for each of the infeasible designs, see Figure 4.

In detail, the results of the feasibility check are collected and the responses for the feasible designs are calculated regarding the demanded load-cases. Above all, the focus is on the response function, which needs to stay continuous. Therefore, the values of the penalized sample should not vary in their dimension. By storing the responses for every sample, an easy penalization of the infeasible designs of the sampling is accomplished. After finalizing the calculation of the responses, the procedure is handling the inferior responses to every infeasible design equally. Furthermore, no resources are spent on the calculation of infeasible designs. The responses are multiplied by an infeasibility factor, which creates a gradient in the sampling. Notably, this causes the infeasible designs to exhibit lower structural performance than the rest of the sampling. In such a way, the infeasible designs remain part of the sampling, supporting the gain of knowledge over the sampling and the infeasibility is graded.

In general, the procedure is applicable for any kind of sampling and optimization

https://doi.org/10.24355/dbbs.084-202111091534-0

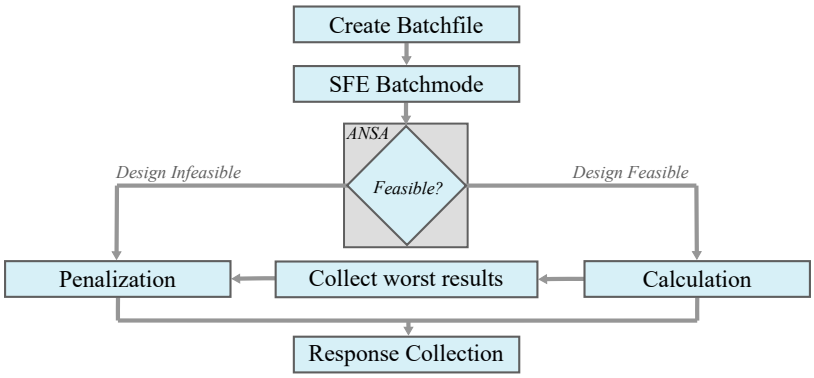


Figure 4. Schematic procedure for an automated feasibility assessment during optimization

Table 1. Penalization Procedure introduced in Figure 4 demonstrated by an example. **The objective is calculated regarding Equation (3), the single constraints by Equation (5) and the overall constraints value is calculated following Equation (7), with the given weights**

	Type						
	Mass in kg	Displacement in mm	Summed Energy *in J*	Intrusion in mm	Infeasibility Factor	Objective in %	Constraints
Weights	1	$\alpha = 0.2$	$\beta = 0.4$	$\gamma = 0.4$			
Sample 1	30.00	1.00	80000	20	0.00	79	0.17
Sample 2	35.00	0.50	60000	25	0.00	92	0.65
Sample 3 (penalized)	36.05	1.03	58252	26	1.03	95	0.91
Sample 4 (penalized)	37.85	1.08	56556	27	1.05	100	1.00

technique. Summarizing, the penalization procedure enables the detection of infeasible designs and helps the optimization routine to mind infeasible areas. As an example of the approach, Table 1 shows the introduced procedure.

Design 1 and 2 are feasible. Thus, the results are calculated in a normal manner. Accordingly, Design 3 and 4 are infeasible and their fitness is penalized. As the infeasibility factor of Design 4 is larger than the one of Design 3, the penalization of the fitness results in a higher fitness value. Besides the detection of the infeasibility, a metric for infeasibility is needed, to create a gradient in the resulting infeasible designs. This measure enables the optimization algorithm to detect the level of infeasibility. Hence, a procedure is introduced, which assesses the surface of the penetration.

Following the approach is introduced briefly and complex geometric changes are processed on the generic b-pillar, which is introduced in Figure 1. The changes in the parameter of the geometry cause the outer shell (grey) to intersect the restricting surface (blue), see Figure 5.

The figure shows the change of two DVs in the lower area of the generic b-pillar. Thus, the intersections occur, when high values for the two DVs are selected.

3.2.2. Infeasibility Factor – Surface Area

The method evaluates the mesh by an element penetration check. The procedure counts the total surface of the elements $A_{E,total}$, the surface of the intersecting elements $A_{E,intersect}$ and the surface of the created groups $A_{E,groups}$. The small groups are created by **the intersecting and separated elements, which are clustered to**

https://doi.org/10.24355/dbbs.084-202111091534-0

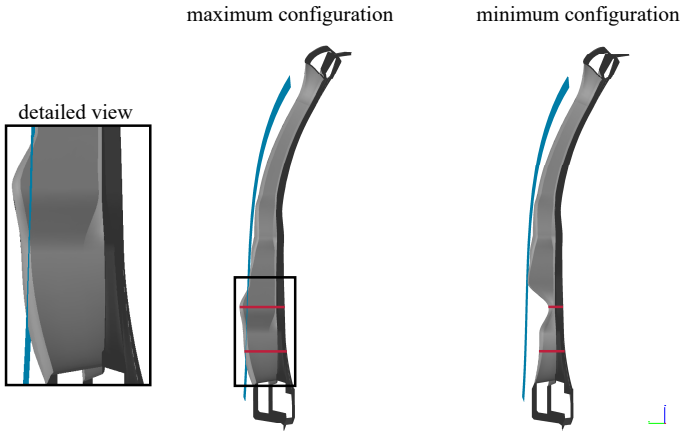


Figure 5. B-Pillar structure (grey) with restricting surface (blue). Maximum configuration of the 2 DVs left-hand side with surface penetrations. Right-hand side, mimium configuration without surface penetrations

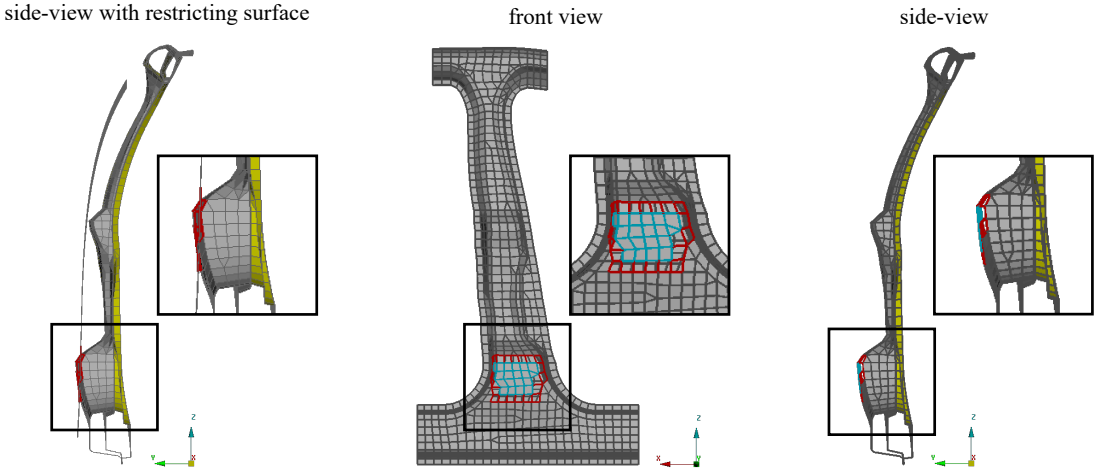


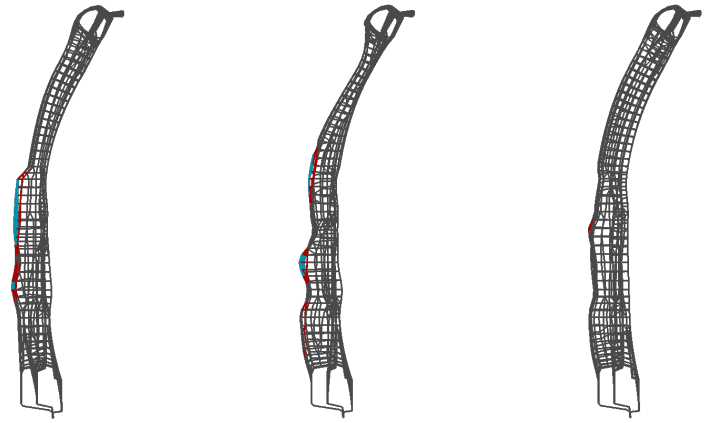
Figure 6. Detection of the intersecting elements in ANSA. Intersecting elements are colored in red and the separated groups are colored in cyan. **The intersecting and separated groups are clustered each in small-groups for every separated group of elements.** The remaining elements of the part/ assembly are colored dark grey

groups. Accordingly, a small-group is arranged for every separated group of elements. Hence, Equation (8) describes the surface infeasibility factor inf_A.

$$inf_A = \frac{A_{E,intersect} + A_{E,groups}}{A_{E,total}} \tag{8}$$

Following Equation (8), a part with less intersecting elements and smaller separated groups has a lower infeasibility factor. If no intersecting elements are detected, the factor turns to zero. In Figure 6 the surface-infeasibility method has been computed for a random configuration of the generic b-pillar. The intersecting elements surface $A_{E,intersect}$ is colored red, while the separated surface of the groups $A_{E,groups}$ is colored in cyan. All remaining elements of the part which are used for the surface $A_{E,total}$ are colored in dark grey.

https://doi.org/10.24355/dbbs.084-202111091534-0



(a) $inf_{total} = 1.0284$ (b) $inf_{total} = 1.0275$ (c) $inf_{total} = 1.0028$

Figure 7. Different DV configurations for the generic b-pillar causing different infeasibility factors. Sorted from a high infeasibility factor inf_{total} (left-hand side) to low factor (right-hand side)

The infeasibility factor inf_A is larger than zero if any infeasibility is detected and equal to zero if no penetration is detected. The surface area of $A_{E,intersect} + A_{E,groups}$ will never be larger than the total surface area $A_{E,total}$. Thus, the factor will always be smaller than one.

$$0 \leq inf_A \leq 1 \tag{9}$$

As the total infeasibility factor inf_{total} needs to be larger than one, the summed value is added by the addend θ. Hence, the following condition needs to be fulfilled $\theta \geq 1$. The complete infeasibility factor is summarized to inf_{total}.

$$inf_{total} = \theta + inf_A \tag{10}$$

This infeasibility factor inf_{total} is applied and tested for three generic examples of the b-pillar. Thus, two large intersections and a small intersecting area are chosen to demonstrate the functionality of the factorization, see Figure 7. The three examples demonstrate the idea for the penalization.

A gradient can be noted for the size of the surface penetrations. The principle approach seems applicable to this simple example. For a better understanding of the approach and to demonstrate the functionality for optimization, the routine is analyzed in the context of an optimization routine.

4. Application of the Procedure for the Structural Optimization of a Generic B-Pillar

Based on the different approaches that were introduced in Section 2.5, a Python-based environment for MDO is developed. The environment is coupling SFE CONCEPT with a DoE and optimization environment. The responses are calculated using a linear and a non-linear (explicit) FE-solver, and thereafter, evaluated for each design. The routine

https://doi.org/10.24355/dbbs.084-202111091534-0

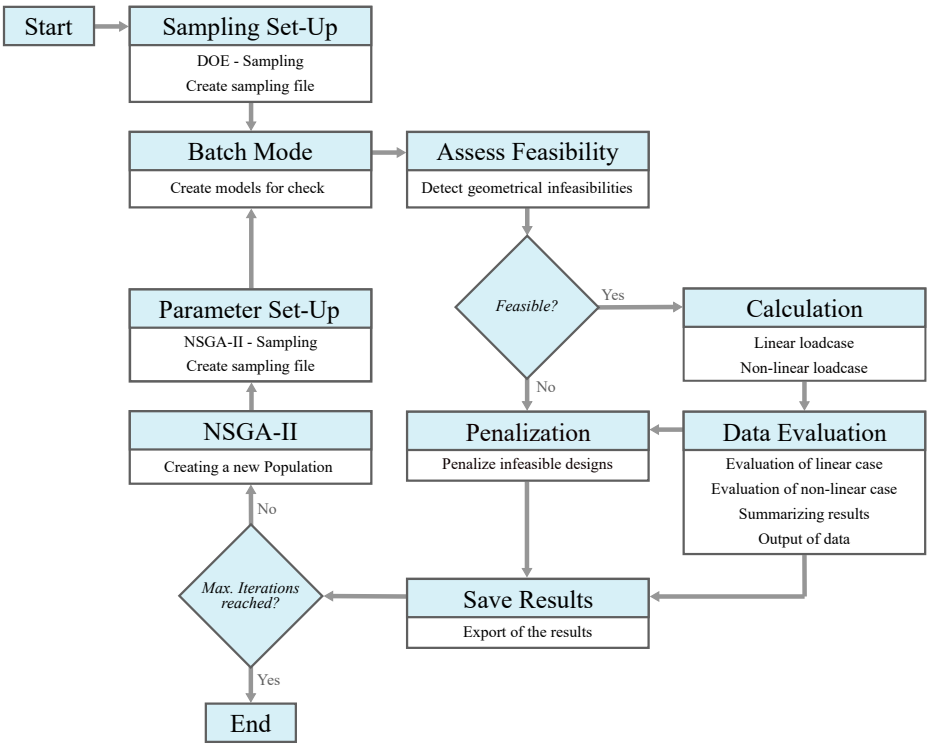


Figure 8. General Process Flow-Chart of the optimization process for shape and size optimization with an NSGA-II including the infeasibility assessment. **Only shape optimization is applied in this example**

is a general procedure for shape and size optimization in SFE CONCEPT. Such approaches have already been implemented for multiple applications, (Duddeck 2008; Hampl and Nammalwar 2011; Rayamajhi, Hunkeler, and Duddeck 2014). Mostly, these implementations use optimization platforms like Heeds (SIEMENS 2020) or ISight (Dassault Systèmes 2020b). In this example, the environment is implemented in Python to make the routine as flexible as possible. The advantage of choosing Python is a large number of implementations and libraries for optimization and machine learning. Initially, the PyDOE-package creates an initial Latin Hypercube Sampling. Due to the simplicity and performance, compare to Section 2.4, an NSGA-II has been implemented in Python, following Deb et al. (2002). Figure 8 shows the process flow-chart with the detection and handling of geometric infeasibility.

The penalization scheme is the core of this approach. In detail, all the designs are evaluated for geometrical feasibility. The calculation of the feasible designs follows the penalization. Finally, the penalization uses the scheme presented in Section 4. For further information on the used software-packages, see [1].

The following chapter tests the performance and applicability of the introduced method. Therefore, the first Section introduces the application for a simple example with two design variables under the infeasibility constraint. Accordingly, the second section presents the results of a more complex optimization task. Both analyses consider a generic b-pillar under two load-cases. Firstly, a linear, torsional load-case and secondly a non-linear side-impact are regarded. The objective and constraint formulation follows Section 3.1. The specification list shows the evaluated and assessed

[1]Used software-packages: ANSA v18.1.0, SFE CONCEPT 2017, NASTRAN 2017 and RADIOSS 2017

https://doi.org/10.24355/dbbs.084-202111091534-0

objectives and constraints:

- Objective: Mass
- Constraints (weighted sum): Static Torsion, Dynamic Intrusion, Absorbed Energy (Internal Energy)
- Geometric Infeasibility Constraint: Yes
- Linear Numeric Evaluations: Nastran (Altair Engineering 2020a)
- Non-linear Numeric Evaluations: Radioss (Altair Engineering 2020b)

The weights for the constraints are applied as given in Table 1. The minimization of the objective and constraints function is the aim of the optimization task. Thus, lower objective- and constraint-function values represent a higher structural performance.

4.1. *Optimization of the Infeasibility and Structural Performance for a two-DVs Example*

The section introduces the results for an NSGA-II optimization for the generic b-pillar structure. The NSGA-II set-up is set to a population size of 40 designs and 10 iterations. In this example, the parameters for the two DVs change the base section of the b-pillar in the lower area, which was introduced in Figure 1. Figure 5 shows the min. and max. configurations of the varied two cross-sections. The parametric changes result in an intersection of the surfaces of the restricting surface for certain combinations of the values. As discussed, the introduced process is automatically penalizing infeasible designs of the sampling.

For this example, the combined metric for infeasibility presented in paragraph 3.2.4 is used. Figure 9 presents the sampling response and the result of the algorithm for the two given DVs. In the figure, the grey symbols represent the initial sampling, and red marks show the designs of the final iteration. The dots represent feasible and the crosses represent infeasible samples. The size of the marker represents the objective value, where a smaller value indicates a higher performance. As the penalization is demonstrated for multi-objective optimization, the example for the normalized constraints violation is not shown, as the penalization produces the same pattern for the objective and the constraints.

The responses **in Figure 9** show, that the infeasible designs have been detected and successfully penalized. The size change of the crossed designs represents the penalization. Accordingly, high values for DV 1 and DV 2 more likely result in high penalization of the created design. This represents the expected behavior, as a larger feasibility factor stands for a larger penetration. The changing sizes of the dots represent the calculated fitness values without any penalization from geometric infeasibility. In the final iteration, the overall fitness of the designs decreased, which is representing a higher structural performance. Besides the evaluation of the performance, the average infeasibility factor of the designs plays an important role, to identify the learning behavior of the algorithm. To underline the learning behaviour of the algorithm with infeasibility constraint, the infeasibility inf_{total} is analyzed more closely in the following section. Additionally, the section gives a more detailed discussion of the performance of the algorithm with and without the infeasibility factor.

https://doi.org/10.24355/dbbs.084-202111091534-0

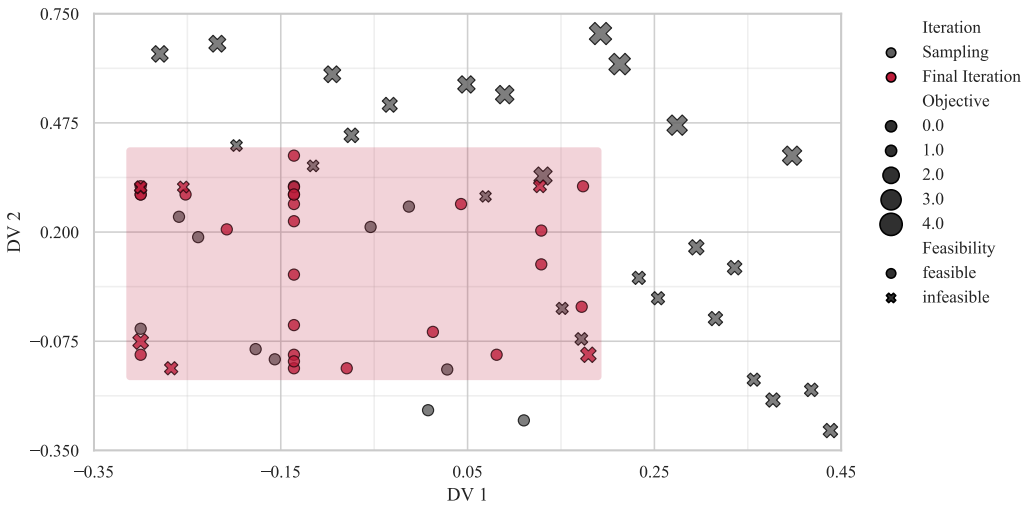


Figure 9. Sampling with a given sampling size of 40 designs and infeasible samples in the sampling plotted for the objective. The penetration of any surface results in the infeasibility of the assembly. Infeasible designs are marked by a (x) and feasible designs by an (o). **The red box shows the converged result of the population based NSGA-II. The DVs are chosen as shown in Figure 5**.

4.2. *Optimization of the Infeasibility and Structural Performance for a 16-DV Example*

Following, an example with 16 DVs is evaluated **to test the routine for more complex tasks**. The outer shell is varied by 8 shape DVs, alike is the stiffener of the b-pillar, **see Figure 1**. **Accordingly, the set-up of the boundary conditions, load-cases and fitness-weights stays unchanged.** The generic b-pillar is optimized for three different configurations. The upper boundaries of the DVs x_{upper} are increased to higher values in the optimization runs (2) and (3), while the optimization run (1) exhibits no intersections with the constraining surface. The configuration is the same as shown in Figure 10, the constraining boundary stays unchanged. As a reference, an initial design (0) is used, which exhibits no changes in the initial geometry and no intersections.

The NSGA-II is set-up for a sampling size of 40 and 20 iterations are processed. In Figure 11, the maximum value $inf_{A,max}$ and the mean value $inf_{A,mean}$ of inf_A are plotted for each generation of a version of the b-pillar with a large surface constraint, which causes large intersecting areas (3). In Figure 11, θ is neglected, so are the feasible designs of each generation. The ratio of the number of infeasible to that of feasible designs of every generation is described with $pop_{inf,ratio}$. The results for the optimization-run (2) with small intersections are neglected, as only the initial sampling evinces five infeasible designs and in the following generations, no infeasible designs are created.

The maximum $inf_{A,max}$ and the mean $inf_{A,mean}$ value decrease over the first three generations. Furthermore, the ratio of infeasible designs $pop_{inf,ratio}$ decreases significantly after the initial sampling. The amount of infeasible designs decreases from 85 % in the sampling to an average of 15 % in the iterations of the NSGA-II algorithm. The optimization algorithm can avoid infeasible areas, as the number of infeasible designs in each generation and the infeasibility factor converge. Besides the convergence of the

https://doi.org/10.24355/dbbs.084-202111091534-0

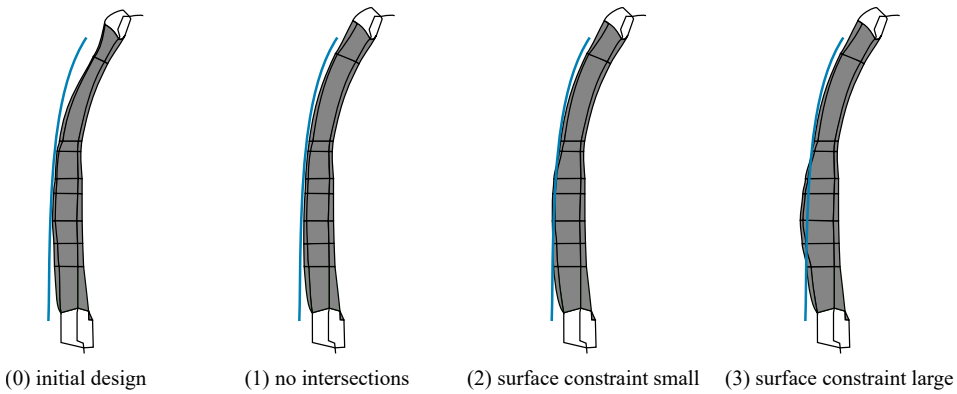


Figure 10. Different maximum configurations of the b-pillar, nominal with no intersections (1), small surface constraints (2) and large surface constraints (3) for the NSGA-II optimizations.

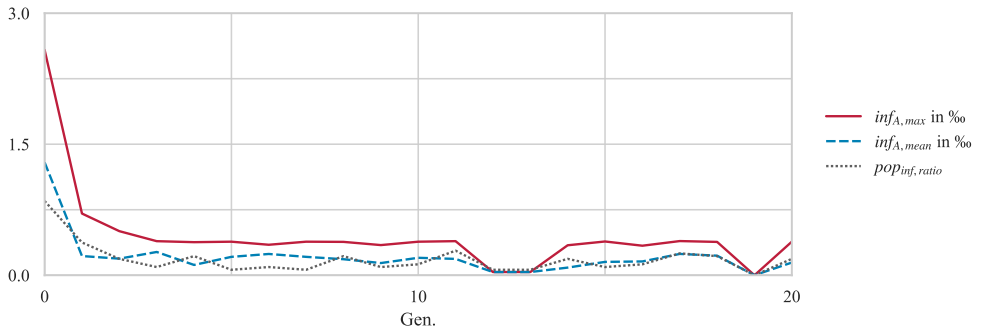


Figure 11. Maximum $inf_{A,max}$, the mean $inf_{A,mean}$ surface infeasibility factor inf_A and the ratio of infeasible designs in the iteration $pop_{inf,ratio}$ for the configuration (3) with 20 iterations and a sampling size of 40, optimized with a NSGA-II

https://doi.org/10.24355/dbbs.084-202111091534-0

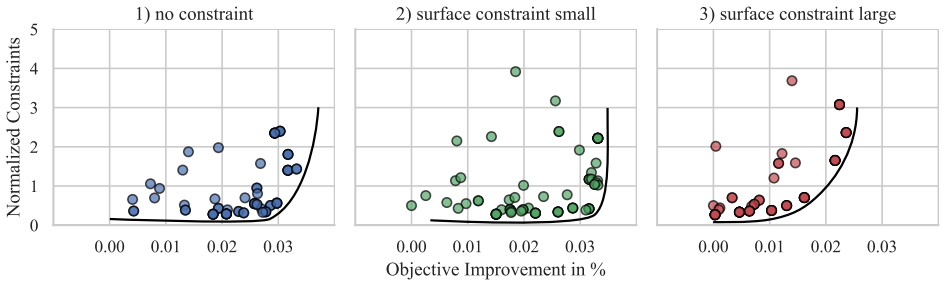


Figure 12. Parents printed for each generation of the NSGA-II compared for the unconstrained b-pillar (1), the constrained b-pillar with surface metric, fitted boundaries and less intersections (2) and the constrained b-pillar with unfitted boundaries and more intersections (3). **Darker elements represent parents which occur in multiple generations of the NSGA-II**

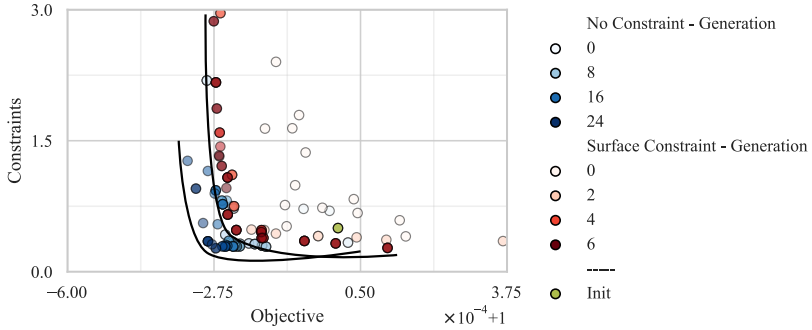


Figure 13. Parents printed for each generation of the NSGA-II compared for the unconstrained b-pillar (blue) and the constrained b-pillar with unfitted boundaries and more intersections (red)

feasibility factor, the structural improvement of the designs needs to be discussed.

Hence, the following paragraph analyzes the improvement of the objective and the constraint violation and the convergence of the NSGA-II for different versions. The higher boundary values cause some of the created designs of the optimization-runs (2) and (3) to be infeasible. The algorithm exhibits different convergence behavior for the runs (1), (2) and (3). Figure 12 shows the Pareto plot of all three configurations.

As seen in Figure 12 the unconstrained optimization with fitted design variable boundaries (1) and the lightly constrained optimization run (2) converges faster than the largely constrained version (3). The performance of the optimization-run (2) can be justified by the small number of initial penetrations in the sampling, which is 17.5 % and no infeasible designs in the following generations.

Hence, the initial sampling size needs to be increased, if large penetration is possible. Figure 13 shows the comparison optimization run (1) and the configuration with large surface constraints (3) for a larger population size of 300 designs.

The optimization results of configuration (3) can be increased by the larger sampling size, Figure 13. The configuration (3) still shows a slightly weaker performance.

5. Discussion of the introduced Method for the Detection of Infeasibilities

The method introduced in Subsection 3.2 enables the detection of infeasibilities and the scaling of the measure. Hence, larger surface penetrations result in a worse perfor-

https://doi.org/10.24355/dbbs.084-202111091534-0

mance than smaller surface penetrations, as demonstrated in Figure 9. This enables the algorithm to gradually handle the infeasibility.

The results from the previous section show that this approach is applicable for multi-objective optimization. The algorithm avoids infeasible areas, as these areas are penalized. The performance of the constrained NSGA-II is comparable to the unconstrained NSGA-II if small surface penetrations are present in the initial sampling. It may underperform compared to the unconstrained NSGA-II if large areas exhibit infeasibilities and the sampling size stays unchanged, see Subsection 4.2. The improvement of the structure for large infeasibilities can be improved to a comparable level by the increase of the sampling size. As a result, a higher need for costly computational evaluations for the mesh-check and numerical calculations arise.

In more detail, the performance which was discussed in Figure 13, is comparable if the sampling size is chosen appropriately large. The reduction of the feasible designs of configuration (3) by 85 % of the initial sampling size can be seen as a drawback for the introduced method for population-based optimization algorithms, see Figure 11. The initial sampling and further generations may shrink critically if the infeasible areas are generous. Vice versa the unconstrained and lightly constrained optimizations have an initial benefit. The detection of the right initial size needs to be addressed by experience or may be handled in future by a classifier algorithm in combination with a resampling.

6. Conclusion and Outlook

The article introduced a new methodology for the detection of geometric infeasibilities in engineering optimization tasks. The approach aims to help during the embodiment design and it fastens the optimization process by the integration and **gradual** judgment of infeasibilities. Existing methodologies have been evaluated and areas of activity are revealed. **Chapter 3** introduced the check for feasibility and an NSGA-II optimization routine, which are both applicable for structural shape and design optimization tasks. For an exemplary application in SFE CONCEPT, the NSGA-II algorithm can avoid infeasible areas and converges. Especially the ability of the algorithm to detect the infeasibility and to avoid these areas is highlighted. External surface representations, like package or design restrictions, can be loaded into the model and are evaluated without costly calculations with a pre-check.

Large surface intersections may cause the NSGA-II to underperform, as the population size can be decreased critically. Hence, the upcoming work focuses on further testing of the feasibility assessment approach. Therefore, different examples will be evaluated. Different optimization routines should be tested, which are not population-based. As well, the performance of the population-based algorithm should be tested with resampling of the initial population. Furthermore, the performance of different surrogate-models and optimization algorithms will be of major interest.

Funding

This work was supported by the Ford University Research Project (URP).

https://doi.org/10.24355/dbbs.084-202111091534-0

References

Altair Engineering, Inc. 2020a. "Altair OptiStruct^{TM} - Optimization-enabled Structural Analysis." Accessed 22.07.2020. https://www.altair.com/optistruct/.

Altair Engineering, Inc. 2020b. "Altair Radioss^{TM} Overview." Accessed 22.07.2020. https://altairhyperworks.com/product/RADIOSS.

Bäckryd, R. D., A.-B. Ryberg, and L. Nilsson. 2017. "Multidisciplinary design optimisation methods for automotive structures." *Int. J. Automot. Mech. Eng.* 14 (1): 4050–4067.

beta cae. 2020. "The advanced CAE pre-processing software for complete model build up." Accessed 22.07.2020. https://www.beta-cae.com/ansa.htm.

Bletzinger, K.-U. 2018. "Shape Optimization." In *Encyclopedia of Computational Mechanics Second Edition*, edited by Erwin Stein, René de Borst, and Thomas J. R. Hughes, Vol. 3, 1–42. Chichester, UK: John Wiley & Sons, Ltd.

Dassault Systèmes. 2020a. "CATIA — SFE CONCEPT - A breakthrough for simulation driven design." Accessed 22.07.2020. https://www.3ds.com/products-services/catia/products/sfe/sfe-concept/.

Dassault Systèmes. 2020b. "ISIGHT & SIMULIA EXECUTION ENGINE." Accessed 22.07.2020. https://www.3ds.com/de/produkte-und-services/simulia/produkte/isight-simulia-execution-engine/.

Deb, K., A. Pratap, S. Agarwal, and T. Meyarivan. 2002. "A fast and elitist multiobjective genetic algorithm: NSGA-II." *IEEE Trans. Evol. Computat.* 6 (2): 182–197.

Duddeck, F. 2008. "Multidisciplinary optimization of car bodies." *Structural and Multidisciplinary Optimization* 35 (4): 375–389.

Duddeck, F. 2016. "Parametric Modeling of Car Body Structures: Enabling Efficient Optimization / Sensitivity and Robustness Analysis for Crashworthiness, NVH, and Multidisciplinary Concept Assessments." White paper on Parametric Modeling of Car Body Structures, TU München, München, Germany.

Feldhusen, J., and K.-H. Grote, eds. 2013. *Pahl/Beitz Konstruktionslehre: Methoden und Anwendung erfolgreicher Produktentwicklung*. 8th ed. Berlin and Heidelberg: Springer Vieweg. http://dx.doi.org/10.1007/978-3-642-29569-0.

Friedrich, H. E. 2013. *Leichtbau in der Fahrzeugtechnik*. ATZ / MTZ-Fachbuch. Wiesbaden and s.l.: Springer Fachmedien Wiesbaden. http://dx.doi.org/10.1007/978-3-8348-2110-2.

Fröhlich, T., S. Kleemann, E. Türck, and T. Vietor. 2017. "Multi-criteria analysis of multi-material lightweight components on a conceptual level of detail." *Proceedings of the 21st International Conference on Engineering Design (ICED17), Vancouver, Canada* (17): 409–418.

Ghaffarimejlej, V., E. Türck, and T. Vietor. 2016. "Finding the best Material Combinations through multi-material joining, using Genetic-Algorithm." *European Conference on Composite Materials, Munich, Germany* (17).

Giesing, J., and J.-F. Barthelemy. 1998. "A summary of industry MDO applications and needs." *Symposium on Multidisciplinary Analysis and Optimization, St. Louis, MO, USA* (7).

Hampl, N., and G. Nammalwar. 2011. "Automated Multi-Disciplinary Optimization (MDO) Process Development and Application." *Weimarer Optimierungs- und Stochastiktage 8.0, Weimar, Germany* (8).

Hilmann, J. 2011. "Correlation of Simulation Models using Concept Modeling." *European Altair Technology Conference - 2011, Bonn, Germany* .

Kiani, M., and A. R. Yildiz. 2016. "A Comparative Study of Non-traditional Methods for Vehicle Crashworthiness and NVH Optimization." *Archives of Computational Methods in Engineering* 23 (4): 723–734.

Messac, A., A. Ismail-Yahaya, and C. A. Mattson. 2003. "The normalized normal constraint method for generating the Pareto frontier." *Structural and Multidisciplinary Optimization* 25 (2): 86–98.

Paas, M. H. J. W., and H. C. van Dijk. 2018. "Multidisciplinary Design Optimization of

https://doi.org/10.24355/dbbs.084-202111091534-0

Body Exterior Structures." *World Congress of Structural and Multidisciplinary Optimisation, Braunschweig, Germany* 41: 17–30.

Rayamajhi, M., S. Hunkeler, and F. Duddeck. 2014. "Geometrical compatibility in structural shape optimisation for crashworthiness." *International Journal of Crashworthiness* 19 (1): 42–56.

Rayamajhi, M., S. Hunkeler, and F. Duddeck. 2015. "Efficient Robust Shape Optimization for Crashworthiness." *World Congress on Structural and Multidisciplinary Optimization, Orlando, Florida, USA* (10).

Ryberg, A.-B., R. Domeij Bäckryd, and L. Nilsson. 2012. "Metamodel-Based Multidisciplinary Design Optimization for Automotive Applications." Technical Report, Linköping University, Linköping, Sweden.

Schumacher, A., M. Seibel, H. Zimmer, and M. Schäfer. 2005. "New optimization strategies for crash design." *LS-DYNA Anwenderforum, Bamberg, Germany* (4).

Shahriari, B., K. Swersky, Z. Wang, R. P. Adams, and N. de Freitas. 2016. "Taking the Human Out of the Loop: A Review of Bayesian Optimization." *Proceedings of the IEEE* 104 (1): 148–175.

SIEMENS. 2020. "Design exploration helps engineers deliver superior performance." Accessed 22.07.2020. https://www.plm.automation.siemens.com/global/de/products/simcenter/simcenter-heeds.html.

Wang, D., and K. Cai. 2018. "Multi-objective crashworthiness optimization of vehicle body using particle swarm algorithm coupled with bacterial foraging algorithm." *Proceedings of the Institution of Mechanical Engineers* 232 (8): 1003–1018.

Winter, J., S. Fiebig, T. Franke, and T. Vietor. 2019. "NURBS-based shape and parameter optimization of structural components with an adaptive amount of control points." *The World Congress of Structural and Multidisciplinary Optimization, Beijing, China* .

Yeniay, Ö. 2005. "Penalty Function Methods for Constrained Optimization with Genetic Algorithms." *Mathematical and Computational Applications* 10 (1): 45–56.

Zimmer, H. 2002. "Erweiterte Knotenfunktionalität im parametrischen Entwurfswerkzeug SFE CONCEPT." *Technical Report, Forschungsvereinigung Automobiltechnik (FAT) e.V.,* (172): 1–30.

Zimmer, H. 2006. "SFE CONCEPT CAE Design: A Key Enabler in Virtual Product & Vehicle Development." *3rd AUTOSIM Workshop, Lisbon, Portugal.* .

https://doi.org/10.24355/dbbs.084-202111091534-0